Perceptron

medium.com/@manveetdn/notes-on-perceptron-padhai-onefourthlabs-course-a-first-course-on-deep-learning-ed0cad95380e

> **Disclaimer: This is notes on "Perceptron" Lesson (PadhAI onefourthlabs course "A First Course on Deep Learning")**

Disadvantage of the MP-Neuron:

y = m*x+b

1. **Boolean input** and **Boolean outputs** only

2. **Linear**(only Linear equation sin. cosine, functions are not possible)

3. **Fixed Slope**(Slope is fixed that might be -1 or so)

4. **Few Intercepts** (only one b (bias))

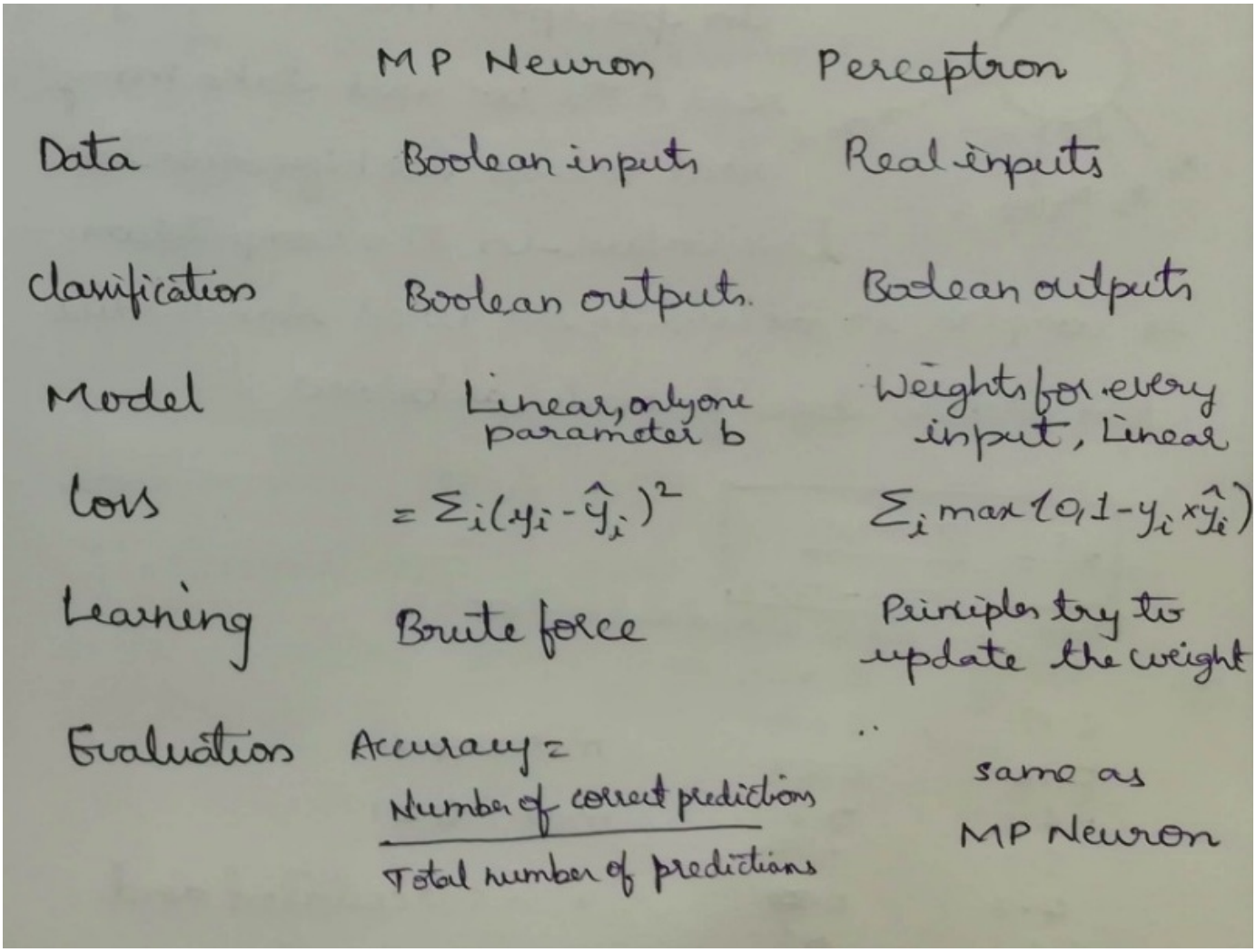
The main reason behind perceptron model to suffice the need of giving non-boolean values as the inputs a better equation to classify more intercepts and a variable slope.

Example : Oil Drilling Case(mining)

It depends on different factors density, salinity,pressure, depth of the oceans surface these all are non boolean values and we want to taken real values and we want to take real values instead of converting values to boolean.

Perceptron:

Perceptron satisfies some of the restrictions faced by the us in MP neurons the main difference are as below mentioned.



Differences between MP-Neuron and Perceptron.

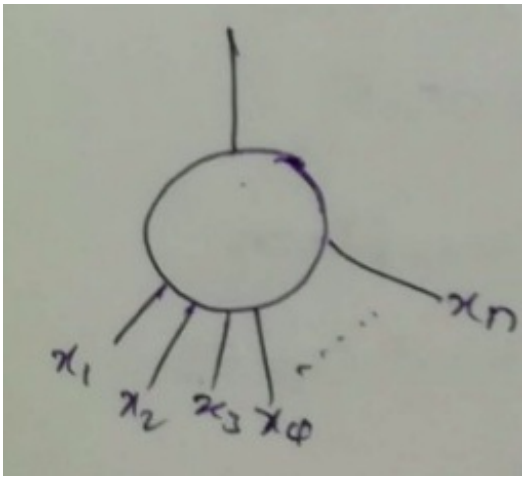
Perceptron:

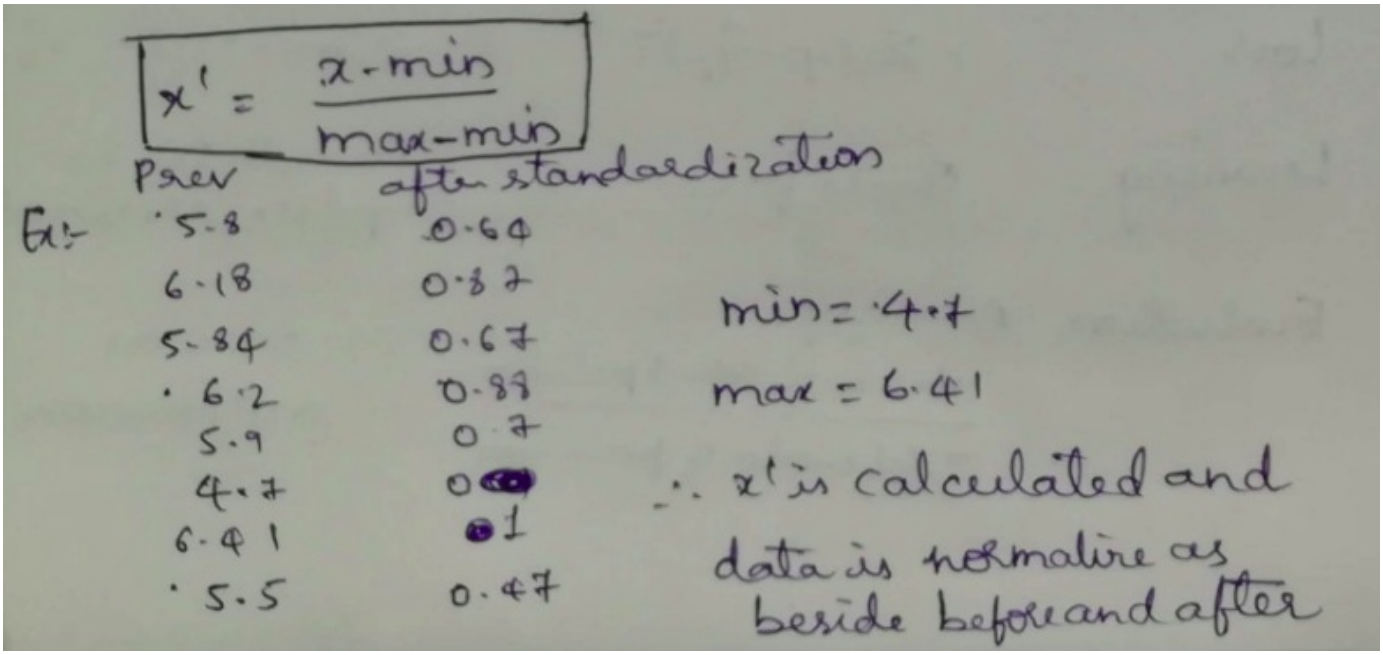
Data and Task:

Perceptron model can even have boolean values but also have real values mainly in the case of this perceptron model.

Perceptron model

In the perceptron model taking real data we will take many real data we will take many real values too high values too low values in this computations is complex so actually we will standardise the data by a formula as below.





Standardisation of data.

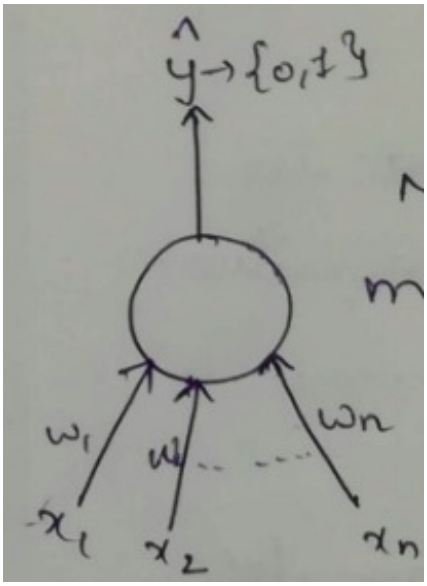
Like this we will standardise the data and we will make data more efficient.

But in perceptron model finally even if the inputs are real the outputs are real the outputs will be boolean.

The Model:

Perceptron.

Its more or less equal to the MP neuron model in diagram the main difference is here all x1,x2,x3,x4,.....,xn are real values and also w1,w2,w3,....wn are the weight assigned to each value.

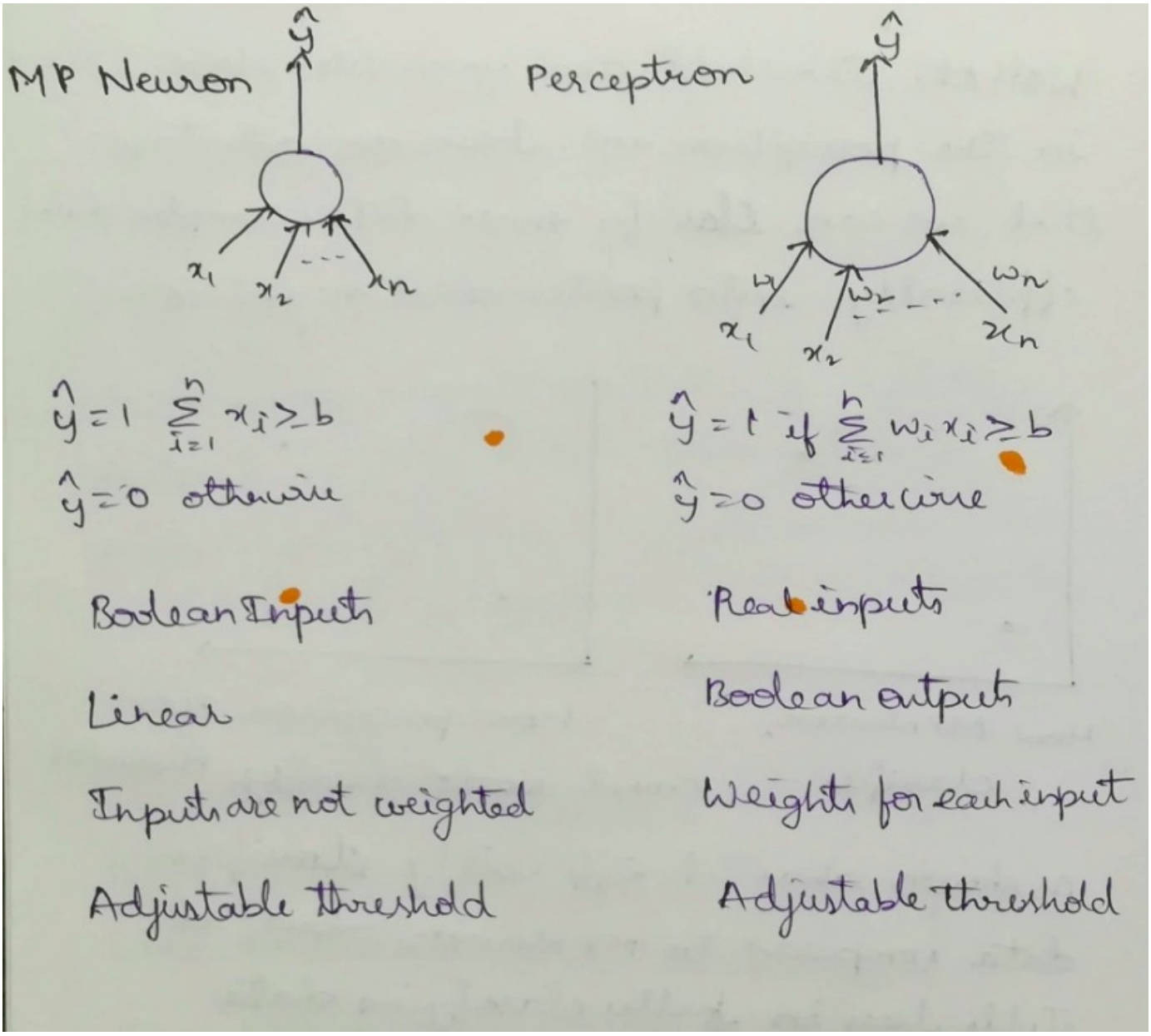


$$\hat{y} = 1 \text{ if } \sum_{i=1}^{n} w_i x_i \geq b$$
$$\hat{y} = 0 \text{ otherwise}$$

Output Condition.

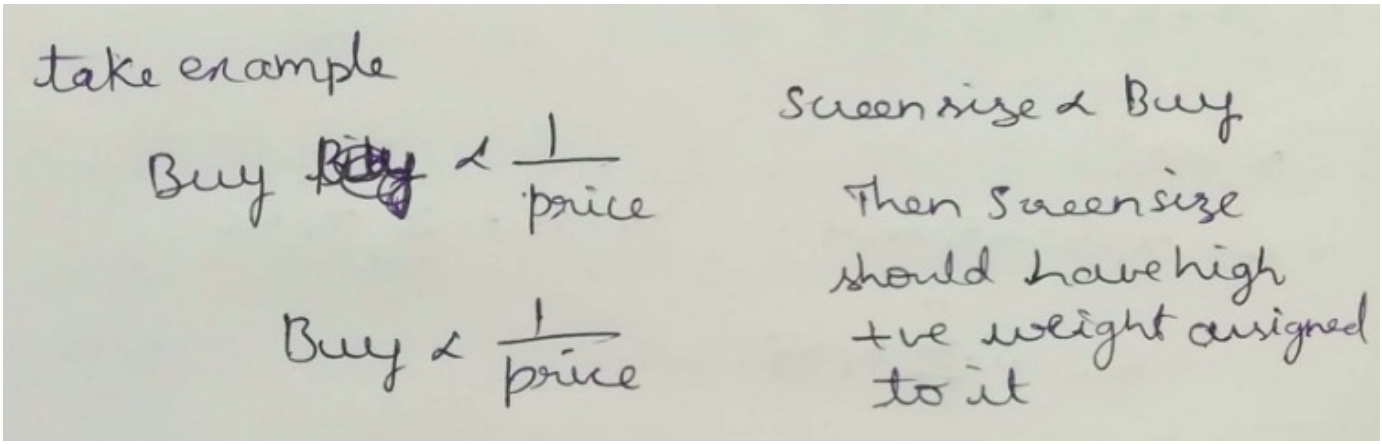
Like that is the summation is greater than b then the output is one else the output is zero.

Difference between MP-Neuron and Perceptron:



Difference between MP-Neuron and Perceptron.

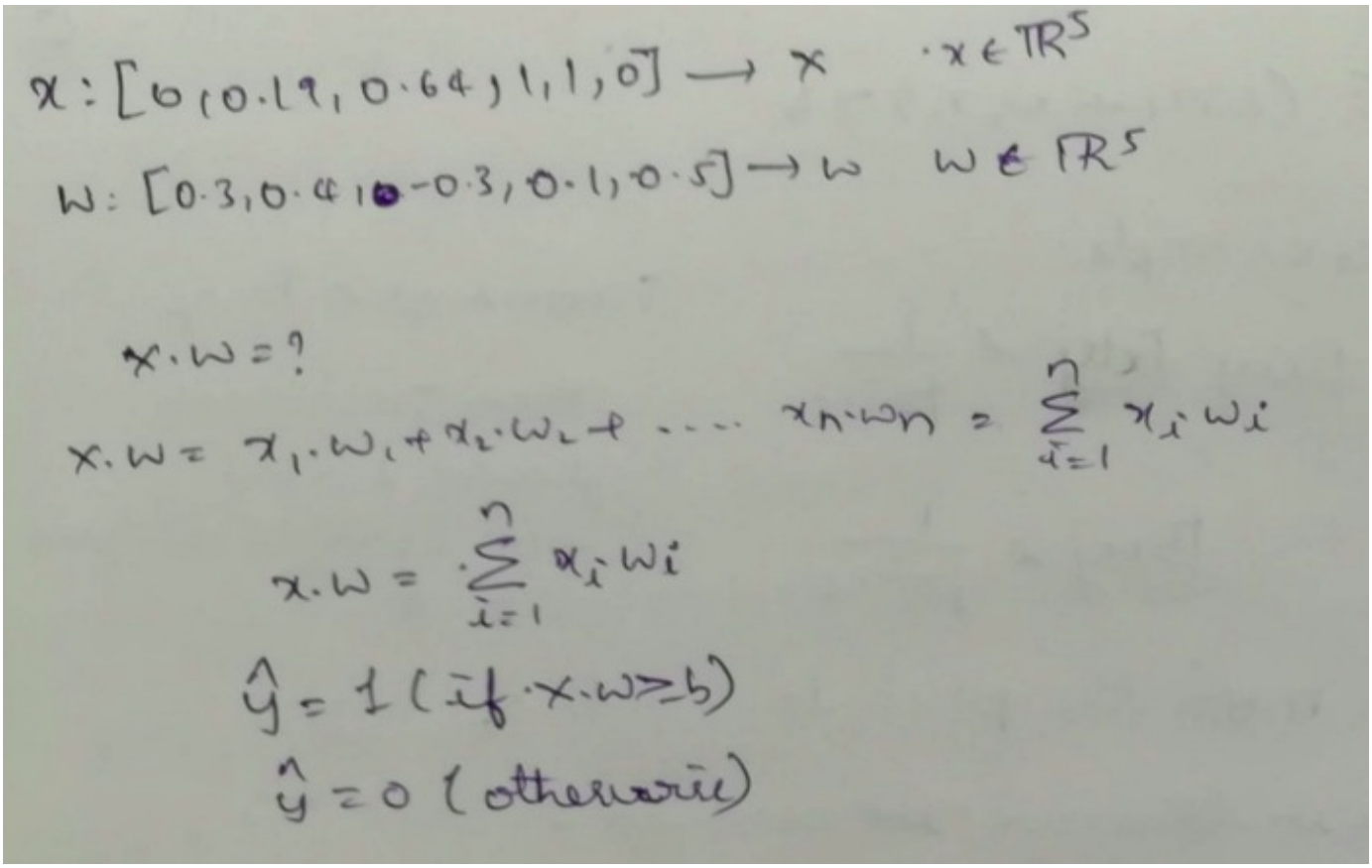
Like that another way of differentiating MP Neuron and the perceptron is



More the price less number of buyers

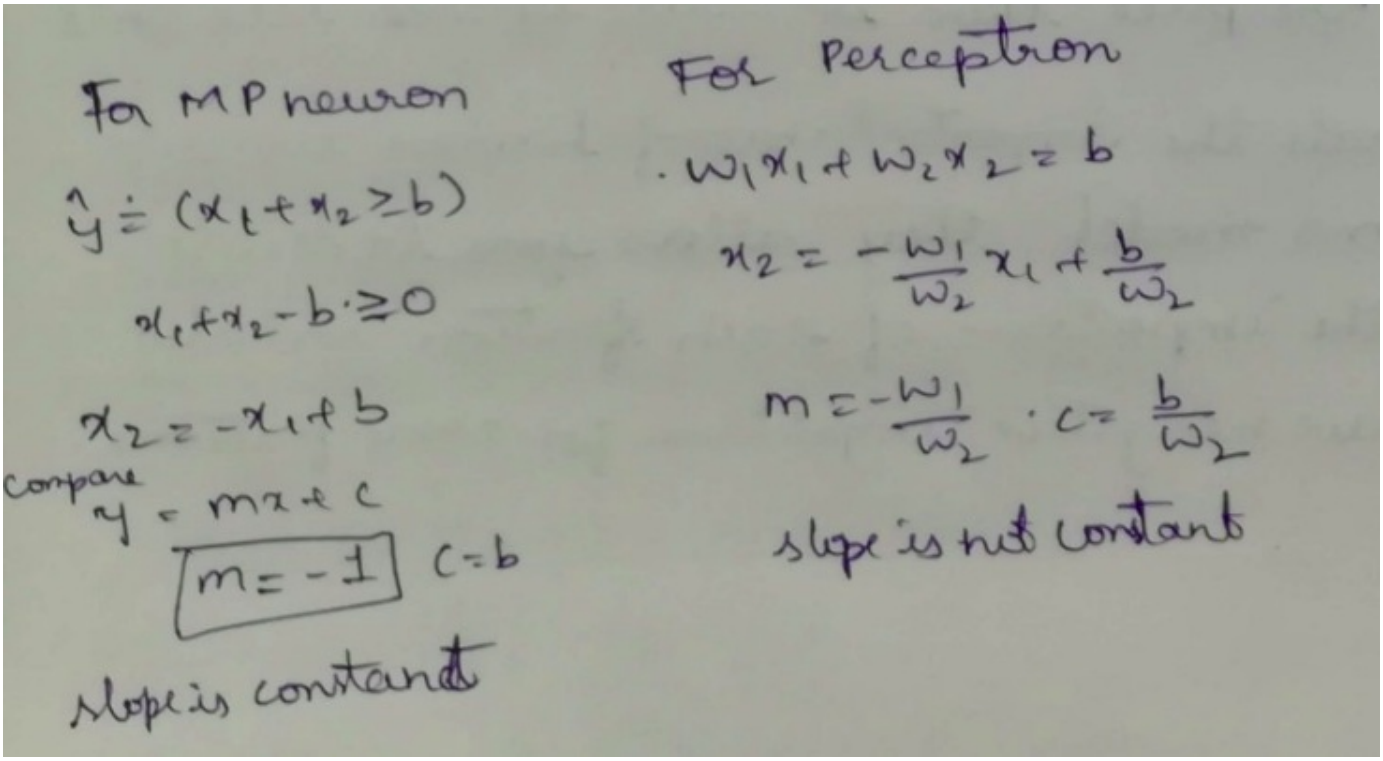
So in this case we need to reduce the weight and assign negative value for the price then we will update like that.

That's the importance of having weights in a model they allow you to decide the importance of each feature and also have a negative importance for each features.



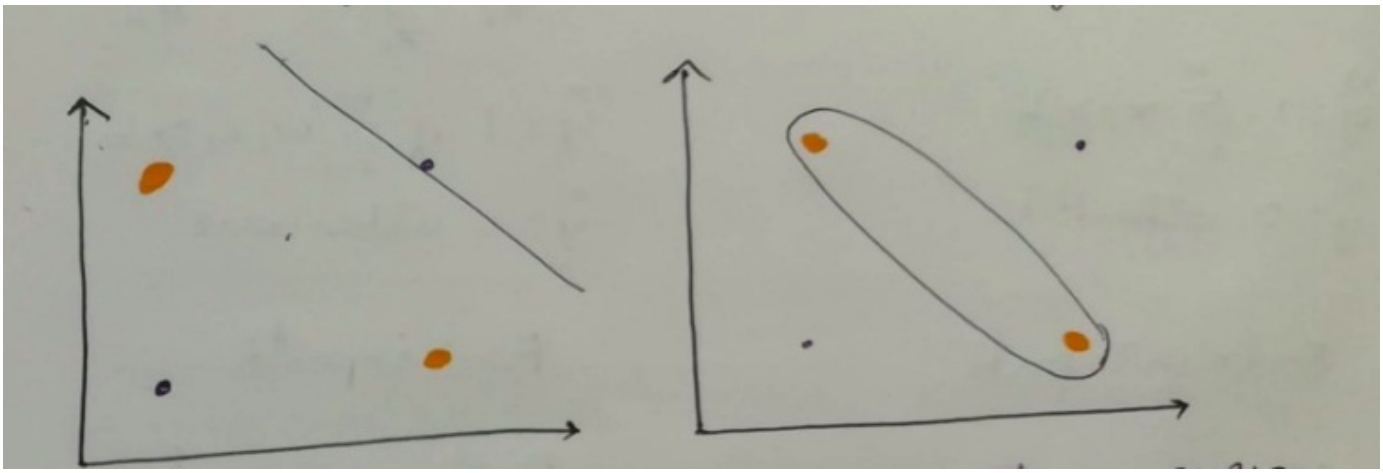
Predict the output of the process

Geometric Interpretation of the model:



This is the geometric interpretation of the of the MP-Neuron and the perceptron.

With the threshold and the variables slope in the perceptron we have an advantage that we can classify more data(complex data) efficiency into positive and negative.



(i)How MP-Neuron classifies, (ii) How perceptron even cannot classify this model

As shown above it efficiently classifies data compared to MP Neuron but it still lags in fully classifying data.

For 3-D shape:

w1*x1 + w1*x2 + w3*x3 — b = 0

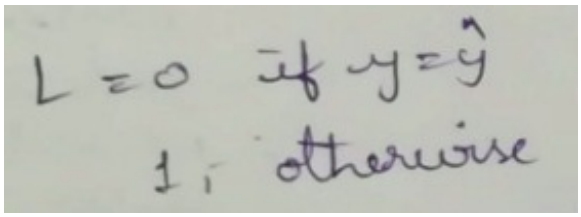
Loss Function:

Here for example purpose if actual output(predicted one) = true output then we will assign a loss else we will assign a loss =1

Perceptron Loss Function

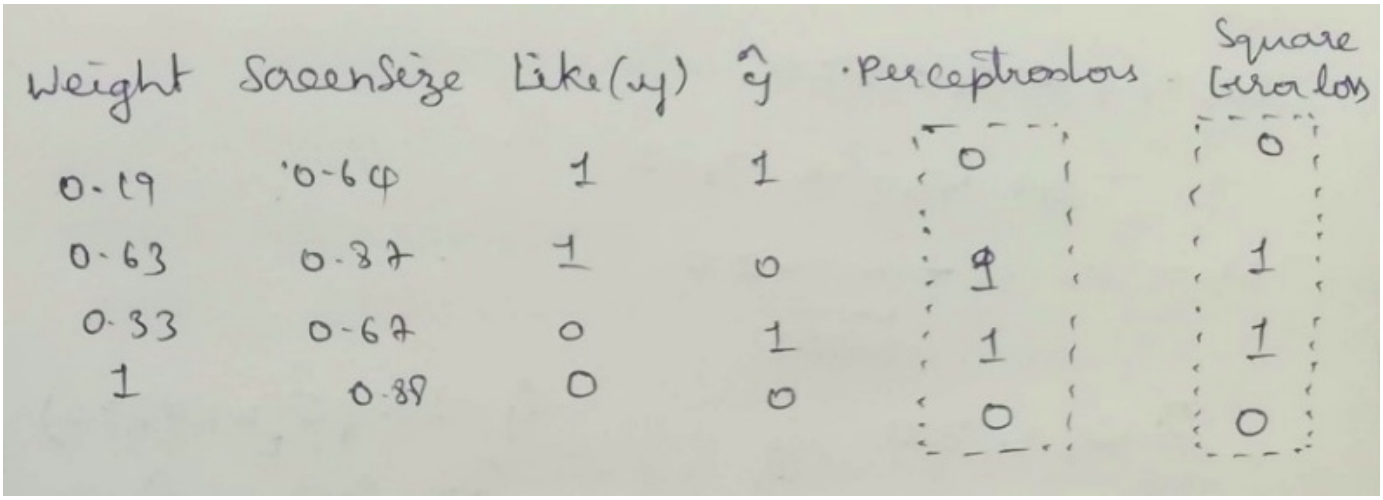
Purpose of the Loss function is to say the model that some correction need to be done.

That is the perceptron loss function.



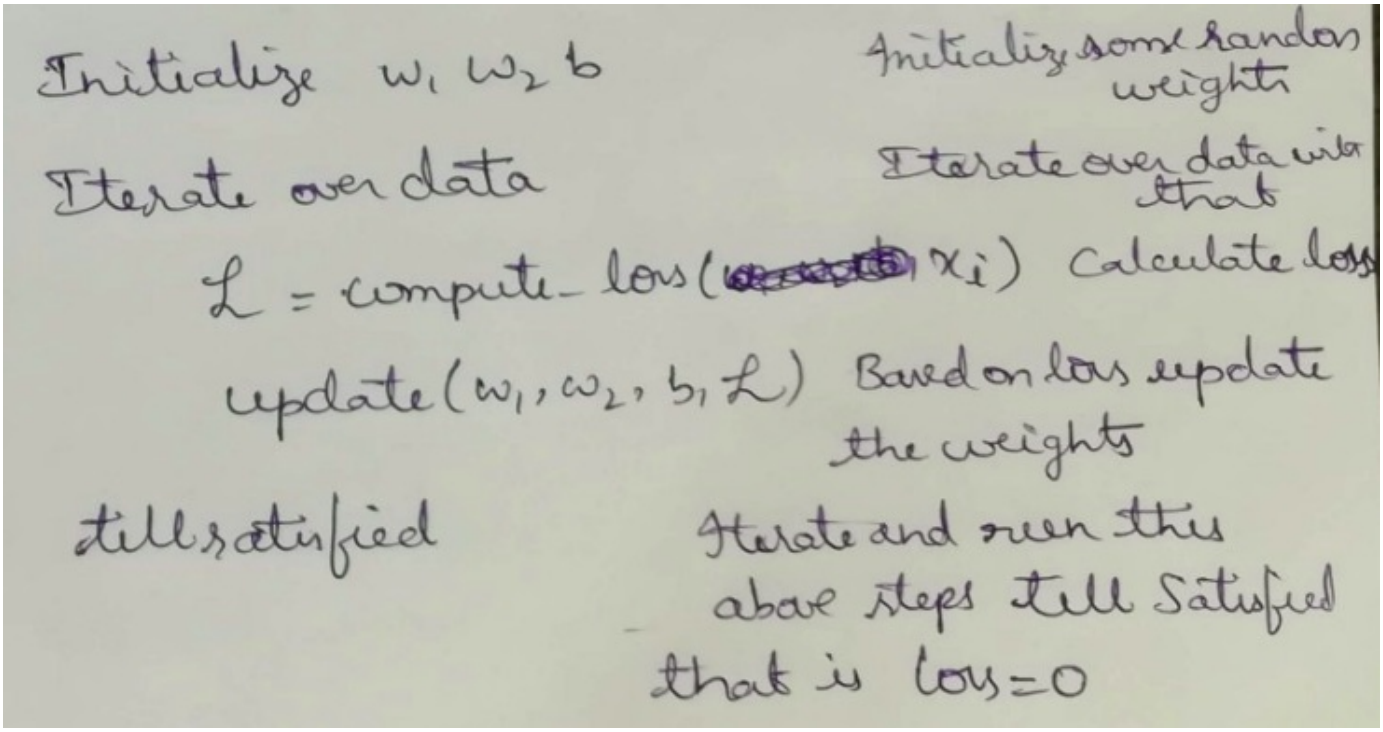
Perceptron loss = Square Error Loss

They both are same When we have boolean outputs.



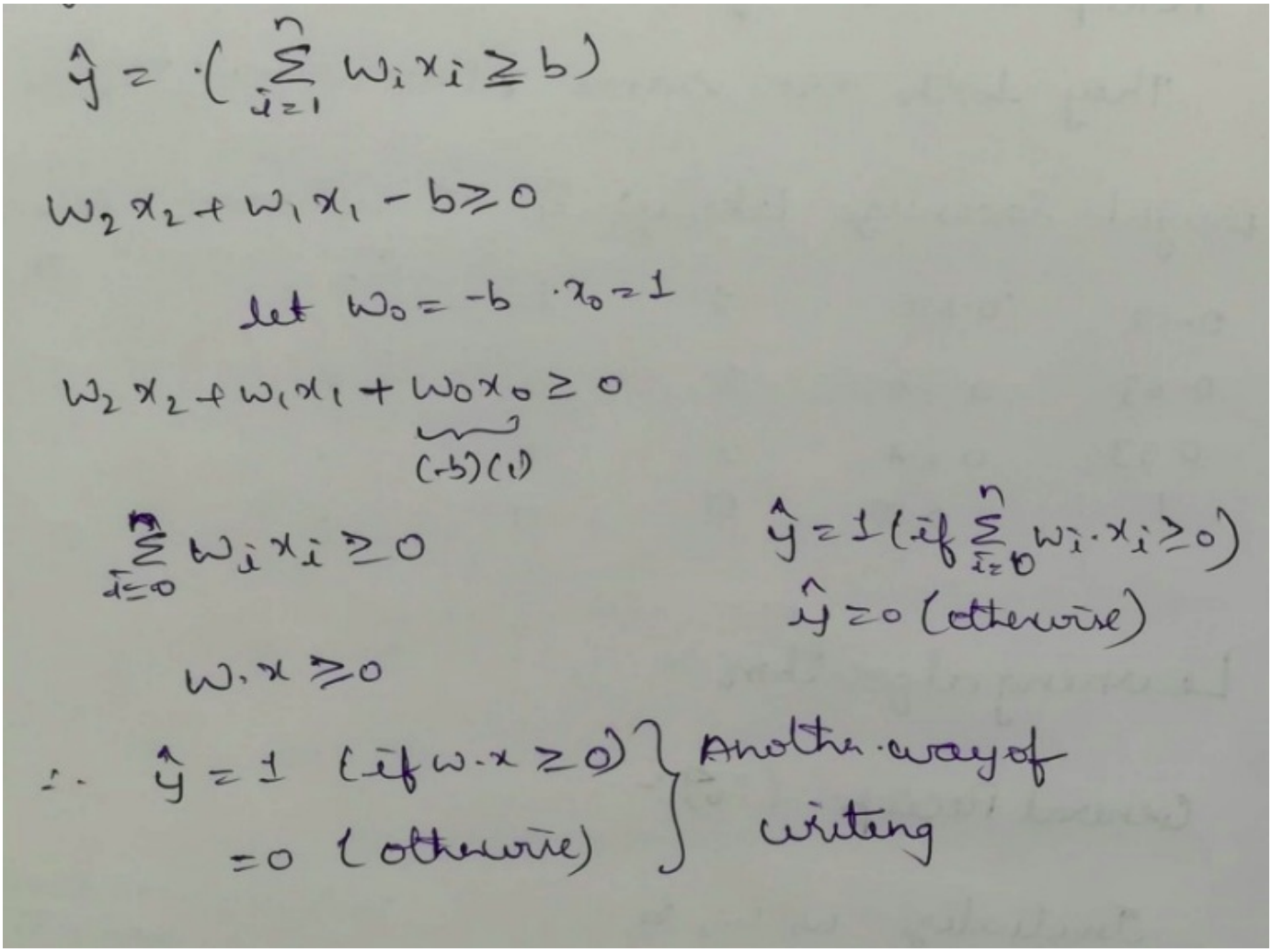
Learning algorithm:

General Recipe :P :

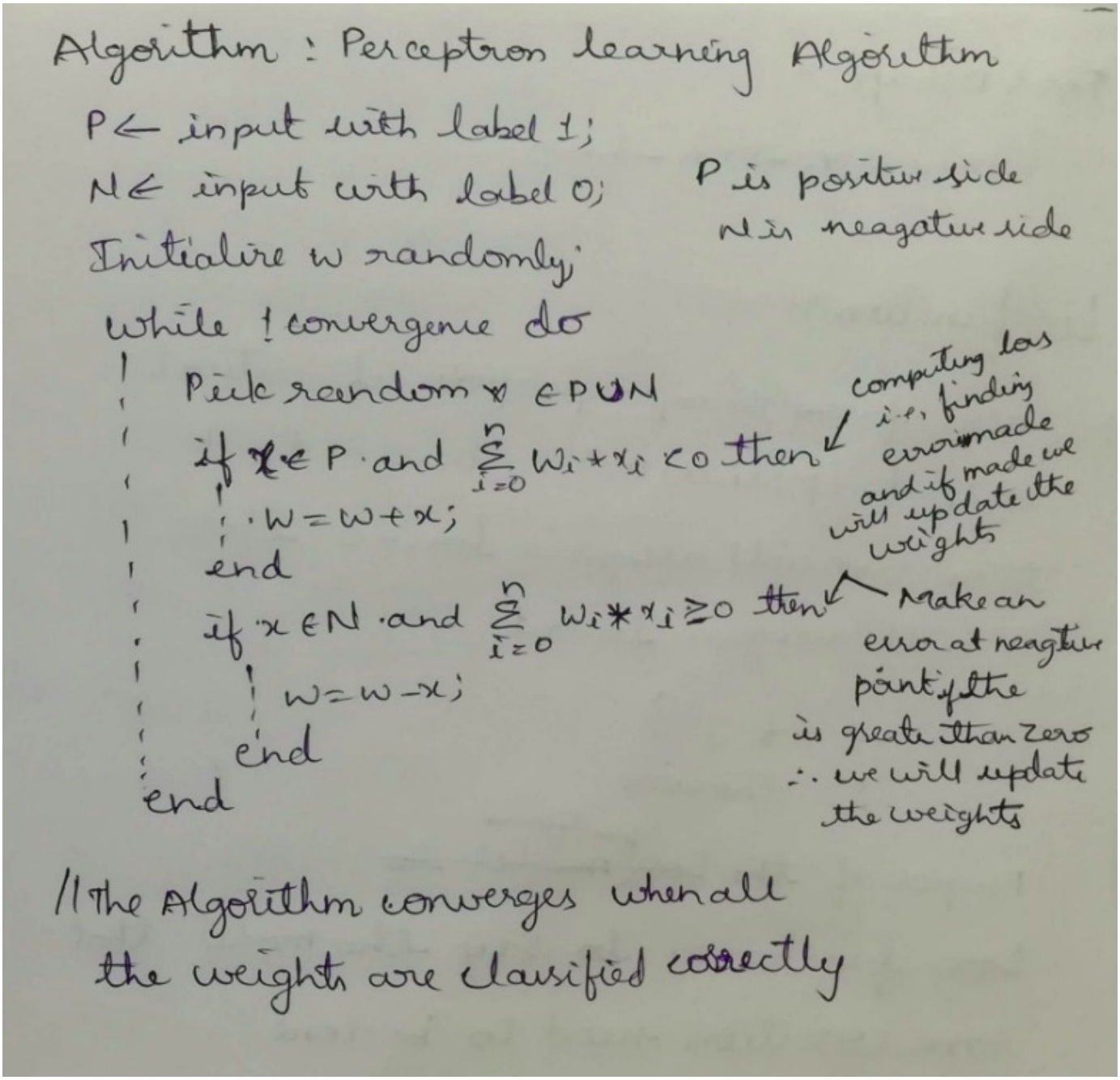


Basic Steps

Algorithm:

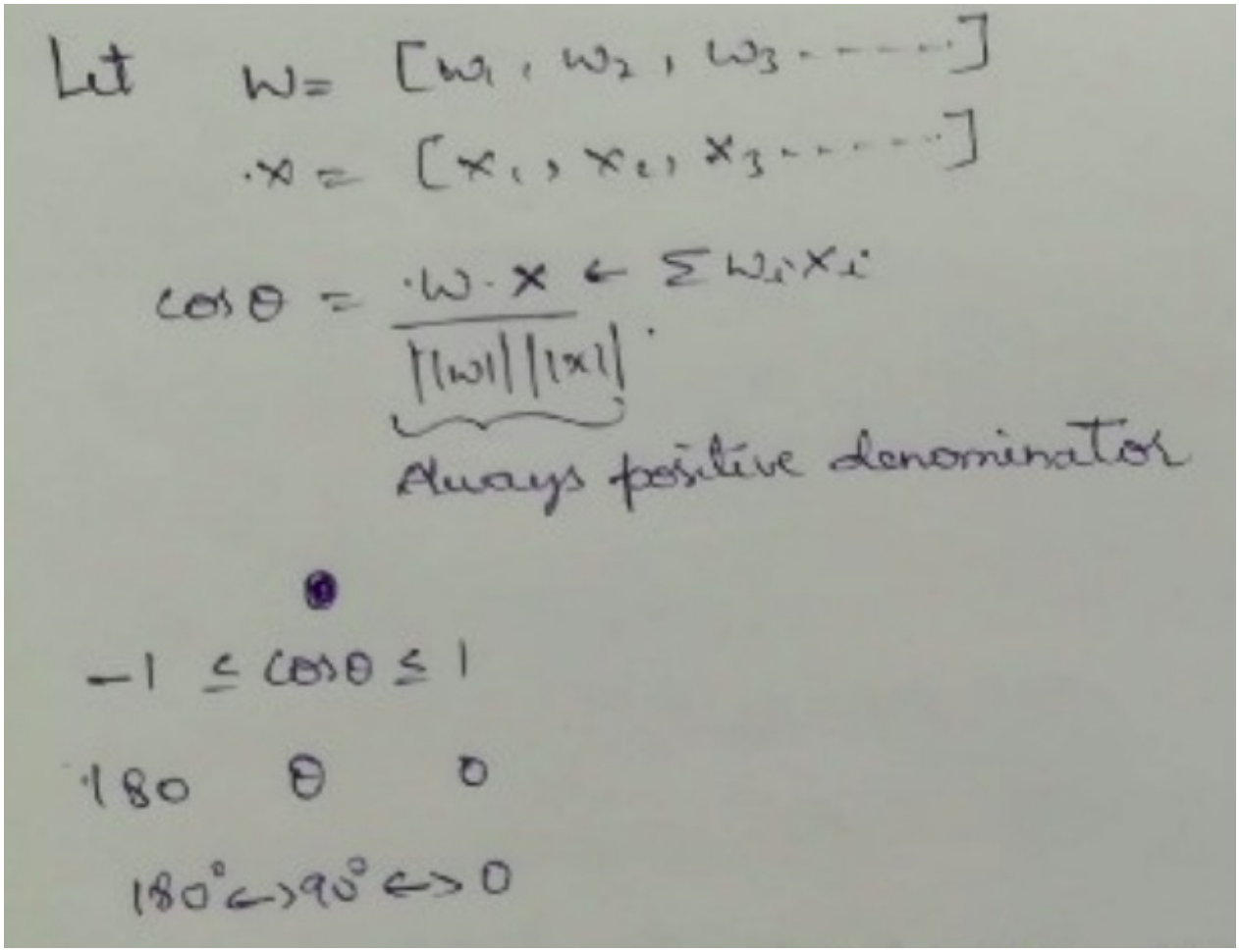


Algorithm what we do actually.



Algorithm of Perceptron

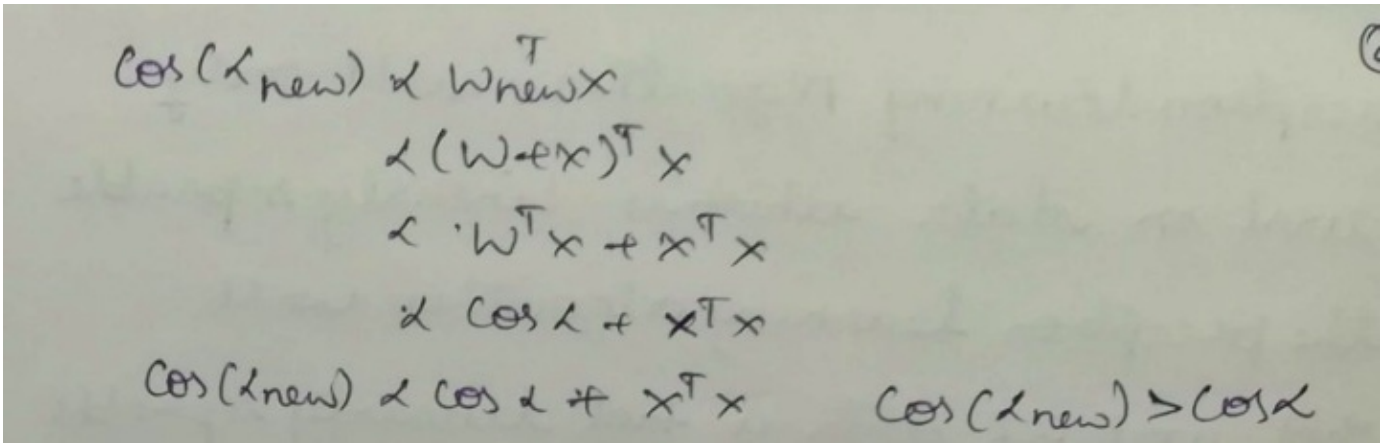
Based on the above Algorithm:



The denominator being positive the sign depends on the dot product w*x

For x belongs to p if w*x<0 (Actually we should have w*x≥0 we check negative case) then it means that the angle (**α**) between this X and teh current W greater than 90 (but we want to be leass than 90) what new angle hapens after update.

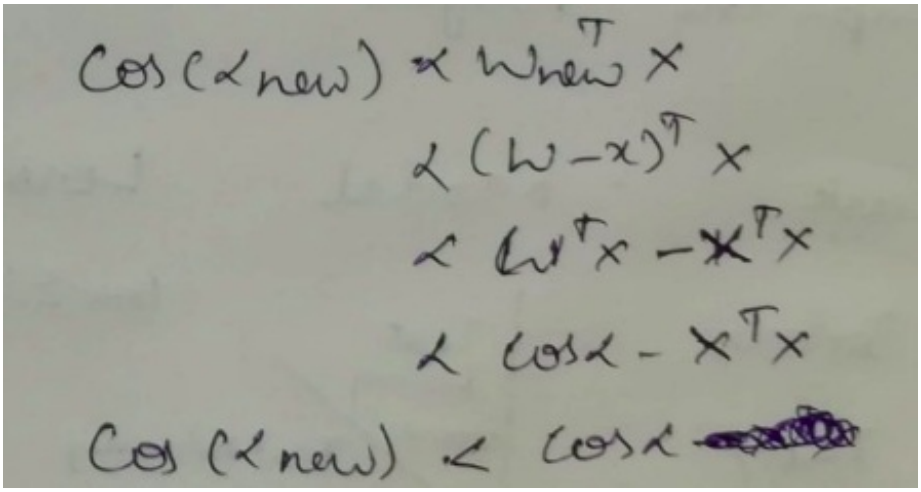
Therefore, w (new) = w + x



Cosine value increases and the angle between vectors decreases.

Similarly, for the negative for x belongs to N if w*x≥0 (Atcually we need it to be w*x<b we see

negative) then it menas that angle (**α**) between this X and the current w is less that 0(but we have want (**α**) to be graeter)

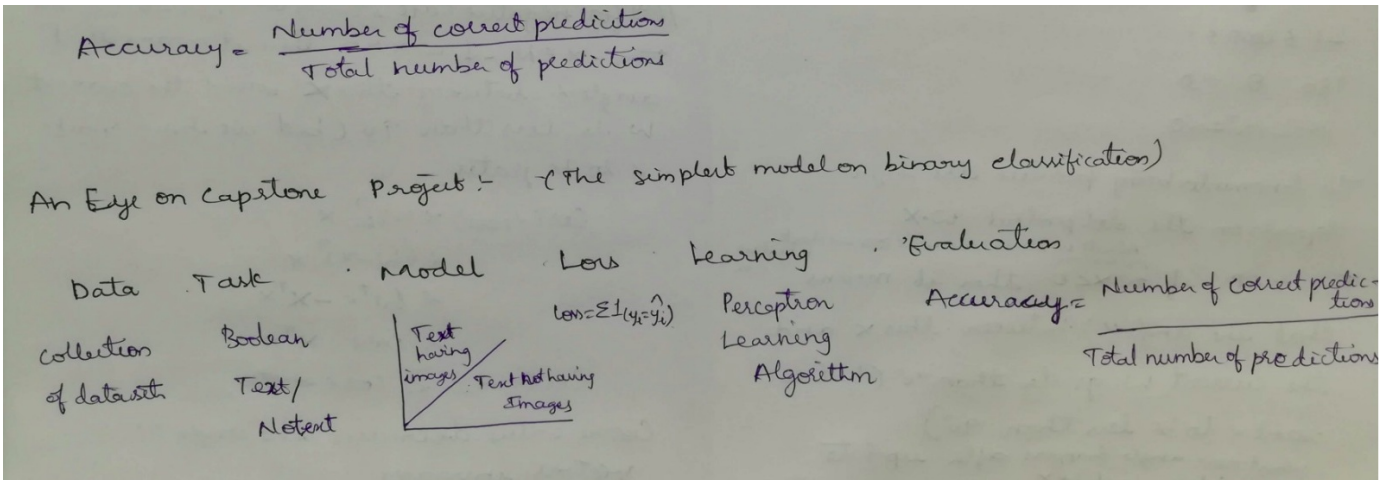


Cosine value decreases the angle between vectors increases.

Perceptron learning Algorithm will only work on data which is linearly separable the perceptron learning algorithm will not working if data is not linearly separable.

Evaluation:

Accuracy formula is the ratio of Number of correct predictions to the total number of predictions.



This is the detailed explanation of about the perceptron model.

This is a small try ,uploading the notes . I believe in "**Sharing knowledge is that best way of developing skills**".Comments will be appreciated. Even small edits can be suggested.

> **Each Applause will be a great encouragement.**

Do follow my medium for more updates……